EXHIBIT 1

EVIDENCE IN SUPPORT OF PLAINTIFFS' OPPOSITION TO DEFENDANT'S MOTION FOR JUDGMENT AS A MATTER OF LAW THAT THE ASSERTED CLAIMS OF THE '115 PATENT ARE NOT INFRINGED AND ARE INVALID OR, IN THE ALTERNATIVE, FOR A NEW TRIAL ON INFRINGEMENT AND VALIDITY

1. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 40:25-43:8; Ex. 22, P-G3 at 35-36. Dr. Hwa C. Torng explained that in his '115 patent, once the dispatch stack determines that instructions are free of data dependencies, it informs the reservations circuit. The reservation circuit then issues multiple instructions at once per clock cycle and can issue the instructions out of order. (*See also* P-G3 at 35 which states "multiple instructions issue at once" and "instructions can issue out of order.") He then discussed an exemplary set of instructions, some of which are data dependency free ("green colored balls) and others of which have data dependencies ("red colored balls").

2. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 46:21-47:10; Ex. 22, P-G3 at 40. Dr. Torng described the reservation circuit as a common component, known since the early 1960s, that arbitrates (*i.e.*, decides) which instructions should be issued, notably multiple and out of order instructions that are dependency free. Specifically, in the context of a slide depicting a reservation circuit and a dispatch stack containing multiple dependency free and out of order instructions (*i.e.*, green colored balls), Dr. Torng explained that the reservation circuit functions as follows: once the dispatch stack indicates that instructions are "ready to go" (*i.e.*, issue), the reservation circuit decides which instructions should be issued by looking to whether the appropriate functional units are available. (*See also* Ex. 22, P-G3 at 35 which states "multiple instructions issue at once" and "instructions can issue out of order.")

3. Ex. 7, 05/21/08 PM Tr. (Dkt. No. 1033) at 286:23-288:9; Ex. 23, P-G5 at 14. Dr. James E. Smith, testifying as a qualified expert, described the reservation circuit consistent with Dr. Torng's explanation. He explained that, when instructions have become dependency free,

the reservation circuit determines whether there is an appropriate available functional unit for each instruction and, if the requisite unit is available, the reservation circuit indicates that the instruction can issue. He sets forth two examples in which this occurs: (1) where two instructions want the same otherwise available functional unit at the same time, but the reservation circuit decides which instruction gets the functional unit and which one will have to wait for the unit; and (2) where a functional unit that the instruction needs is "busy," the reservation circuit will hold the instruction from issuing until the unit is available.

4. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 13:4-16:15; Ex. 23, P-G5 at 22. Dr. Smith further described the reservation circuit consistent with Dr. Torng's explanation. Again, in the context of a slide depicting a reservation circuit and a dispatch stack containing multiple dependency free and out of order instructions (*i.e.*, green colored balls), Dr. Smith explained that once instructions are data dependency free, the reservation circuit determines whether there is an available functional unit for each such instruction and, if so, issues multiple and out of order instructions simultaneously.

5. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 68:4-7. As to claims 1 and 14, Dr. Smith explained that "means for detecting equals the dispatch stack, [and] means for issuing equals reservation circuit."

6. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 65:5-23; Ex. 23, P-G5 at 53-55: Dr. Smith explained that, based on his review of the ERS (P-355) and the schematics, the PA-8000 family processors have arbitration logic

7. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 65:5-24-66:22, 67:14-68:7; Ex. 23, P-G5 at 52-55, 61: Dr. Smith described the arbitration logic in the PA-8000 family processors with reference to P-G5 slides 53-55. He explained that, "the IRB initially determines if instructions are free of dependencies and that's indicated by these little arrows" (slide 53) showing two multiple and out of order dependency free instructions ready to issue. He

explained that the arbitration logic gets a signal that these two instructions need multipliers (*i.e.*, functional units) and determines whether a multiplier is available. He explained that, if only one multiplier is available, the arbitration logic decides which instruction will get the multiplier. He also explained a second instance in which the arbitration logic in the IRB is used; namely where an instruction needs a divide-square root functional unit, but that functional unit is "busy with some previous instruction" and thus not available. So, the instruction has to wait until arbitration logic determines that the divide-square root functional unit is available. Dr. Smith also referenced slide 61 which states, in part: "Arbitration Logic = Reservation Circuit (Claims 1 and 14)."

8. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 85:12-89:23; Ex. 23, P-G5 at 92-100. With reference to the ERS (P-355) and schematics (P-801, 813, 819), Dr. Smith testified that the arbitration logic in the IRB is the same or at least equivalent to the reservation circuit recited in claims 1 and 14. Dr. Smith first summarized his earlier testimony about the purpose of the arbitration logic, stating that it "determines if there are available functional units for an instruction when its ready [to issue]." (85:20-25.) In this regard, slide 92 captures an animation of the IRB detecting dependency free instructions (green colored balls) and its accompanying arbitration logic performing multiple and out of order instruction issuance. As with the dispatch stack, slide 92 indicates that as to the IRB with its accompanying logic: (1) "Multiple Instructions Issue at Once" and (2) "Instructions Can Be Issued Out of Order." With reference to the ERS and schematics, he elaborated again on the at least two situations when the launch arbitration logic acts to issue dependency free instructions (busy divide-square root unit, and two instructions wanting the same sole functional unit at the same time). (86:1-88:20.) He identified the specific input and output signals for the arbitration logic that (1) indicate whether an instruction is free of data dependencies; (2) indicate what functional unit is desired; and (3) tell the instruction to issue to the appropriate functional unit. (88:11-89:23.)

9. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 94:14-99:1; Ex. 23, P-G5 at 103, 105-107. Dr. Smith testified that the IRB issues multiple and out of order instructions in a single clock (a/k/a processor) cycle and that the launch arbitration logic in the IRB is responsible for doing so. In so doing, he explained that the IRB meet the claim term "single clock (a/k/a processor) cycle as construed by the Court (97:9-99:1), relying in part on the ERS (P-355) and the Kumar 1997 IEEE article (P-315).

10. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 234:20-235:25, 236:16-238:16; Ex. 20, Swartzlander Dep. Designation at 316:9-13, 316:15-18. Dr. Earl Swartzlander, expert for HP, acknowledged that Dr. Smith testified that the arbitration logic in the IRB corresponded to the reservation circuit in claims 1 and 14 of the '115 patent. He conceded that, in arriving at his opinion, Dr. Smith discussed schematics describing this arbitration logic and talked about the various signals that are input and output into the arbitration logic. (234:20-235:14.) Dr. Swartzlander admitted that the IRB in the PA-8000 family processors issues multiple and out of order instructions in a single processor cycle (a/k/a single clock cycle). (235:18-25.) After denying that the processors met the Court's construction of then term single processor cycle (a/k/a single clock cycle), he was then shown an article authored by Mr. Kumar, Dr. Swartzlander was then impeached with his own prior deposition testimony in which he had admitted that a fair interpretation of this article was that the term as construed was met. Mr. Kumar is a co-author of the chapter in the ERS which describes how the IRB works. (236:4-238:16 and Swartzlander Deposition Designation at 316:9-13, 316:15-18.)

11. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 84:9-13. Mr. Jonathan Lotz testified that the IRB "absolutely" has to efficiently detect whether instructions have any data dependencies in order to do its job of quick and efficient multiple and out of order instruction issuance.

12. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 107:19-24. In the context of the IRB for the PA-8000 family processors, Mr. Lotz testified that launch means issue.

13. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 11:11-12:5. Mr. Gregg Lesartre testified (11:11-23) that, when then IRB has multiple instructions that are data dependency free and ready to launch (*i.e.*, issue), the launch arbitration logic in the IRB selects which instructions to launch and to the functional units. Mr. Lesartre further testified (11:24-12:1) that this selections process is referred to as arbitration and that (12:2-5) it is a "standard function that's commonly used" in processors.

14. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 68:4-7; Ex. 23, P-G5 at 57-58, 61. As to claims 1 and 14, Dr. Smith explained that "means for detecting equals the dispatch stack, [and] means for issuing equals reservation circuit."

15. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 67:2-69:18; Ex. 23, P-G5 at 57-63. Dr. Smith testified that the IRB contains all of the elements of the dispatch stack in the construed claims (claims 1, 6, 14-15, and 18). He also testified that the PA-8000 family processors infringe all of these claims.

16. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 67:2-85:11; Ex. 23, P-G5 at 64-89. With reference to the ERS (P-355) and HP schematics (*e.g.*, P-790) and an extensive set of slides and his earlier testimony concerning the IRB's components and operations, Dr. Smith explained in detail that the IRB contains all of the elements of the dispatch stack, literally or at least equivalently.

17. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 23:1-24:10, 24:14-26:20, 27:20-23; Ex. 23, P-G5 at 25-31. With reference to HP's ERS for the PA-8000 family processors and in particular a block diagram of the processors, Dr. Smith explained, inter alia, that the IRB holds computer instructions that are issued to functional units. He explained that the processor has

architected registers (a/k./a general registers) and rename registers which hold values for the instructions.

18. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 05/22/08 Tr. 46:3-20. Dr. Smith explained that renaming occurs prior to the time that instructions in the instruction issuance unit are considered for issuance.

19. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 48:5-23, 49:7-50:7; Ex. 23, P-G5 at 44-45. With reference to the ERS (P-355), Dr. Smith identified the elements of each slot (a/k/a cell) in the IRB, including the source1 and source2 tags and the control element which deals with detection of data dependencies among instructions.

20. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 51:3-25, 56:11-57:17; Ex. 23, P-G5 at 46-49. With respect to the destination register ("D") for each instruction, Dr. Smith explained (*e.g.*, 51:10-16, 56:15-18) that an instruction residing in the IRB will write its result to a rename register. (*i.e.*, the rename register will hold the instructions computed result). So, in light of renaming the destination register specified by the instruction is a rename register. With reference to the ERS (P-355) and testimony of Mr. Lotz (slides 48-49), Dr. Smith explained that there is one and only one specific rename register per slot in the IRB and that the number of a given rename register is identical to the number of the slot in which the instruction is located (*e.g.*, rename register zero is used with the instruction in slot zero).

21. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 51:3-25; Ex. 23, P-G5 at 46-49. With respect to the source1 and source2 tags in each slot in the IRB, Dr. Smith explained that they have two characteristics: (1) each tag identifies the particular slot numbers of an instruction in the IRB, which slot in turn corresponds to a particular rename register; and (2) each tag identifies, by number, the rename register that produces the source value for another instruction. As such, Dr. Smith explains that, in light of renaming S1 and S2 specify particular rename registers.

EXHIBIT 1

22. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 52:1-21; Ex. 23, P-G5 at 46-49. Dr. Smith explained that, in light of renaming, the value that will be supplied to a source register (*e.g.*, S1) in the instruction comes from either a unrenamed register or a rename register, but the value itself does not change.

23. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 55:3-6. Dr. Smith explained that instructions in the IRB include instructions in OP S1 S2 D format.

24. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 55:7-56:1; Ex. 23, P-G5 at 44-46. Dr. Smith testified that, for ease of explanation to the jury, he rearranged certain of the element of each slot in the IRB, but that he did not in any way change anything about the elements of each slot in the IRB. He explained that he faithfully followed the ERS's (P-355) depiction of these elements.

25. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 56:2-24; Ex. 23, P-G5 T 44-46. With respect to the source1 and source2 tags in each slot in the IRB, Dr. Smith, consistent with his earlier testimony (51:18-25) explained again that they "represent two things really:" (1) each tag identifies the particular slot numbers of an instruction in the IRB, which slot in turn corresponds to a particular rename register; and (2) each tag identifies, by number, the rename register that "going to hold the value for that [other instruction's] source operand. As such, Dr. Smith explains that, in light of renaming S1 and S2 identifies (*i.e.*, specifies) a particular rename register.

26. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 56:25-57:12; Ex. 23, P-G5 at 44-46. Dr. Smith explained that his testimony did not in any way change anything shown in the schematics or ERS for the PA-8000 processors and that the ERS (P-355) was consistent with his explanation.

27. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 57:13-17. Dr. Smith testified that there is exactly one and only one rename register to hold a result specified by a particular slot.

28. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 59:15-64:17, 65:5-18, 65:24-67:1; Ex. 23, P-G5 at 52-55. Using a sample set of instructions loaded in the IRB, Dr. Smith went through a detailed example of how the IRB detects whether instructions are free of data dependencies and works with the IRB's arbitration logic to simultaneously issue (a/k/a launch) multiple and out of order instructions per clock cycle that the IRB has determined are free of essential data dependencies (*e.g.*, shown by the two out of order instructions with the blue arrows in slide 53). Among other things, he explained that (1) the specified source registers (S1 and S2) for the instructions can be unrenamed or renamed registers (61:7-62:15; 63:24-64:17); (2) the slot numbers in the IRB correspond to a particular rename register (*e.g.*, slot number one corresponds to rename register, R1 (61:7-13)); (3) the S1D and S2D flags in the IRB performed the recited function and delineated the recited function (62:3-9 and 63:6-13); (4) a value of one in a given flag indicate that there is a one essential data dependency and a value of zero indicates that there are not any essential data dependencies (61:19-63:13); (5) the value of one in each such flag is decremented to zero with the result that the instruction is free to launch as long as there is an available functional unit (*id.*); and (6) the specified destination registers (D) for instructions residing in the IRB are rename registers (61:7-18). Later, Dr. Smith further reiterated that "the IRB initially determines if instructions are free of data dependencies." (65:24-66:8 and P-G5 slides 53-55.) Then, with reference to slides depicting an IRB, he explained that such dependency free instructions ("that's indicated by these little arrows") and the instructions are simultaneously issued to the functional units with the arbitration logic. (*Id.*) Signals read the AL indicating that the instructions are "ready" to issue. Thus, Dr. Smith determined the recited detecting function and issuance (a/k/a/ launch) function.

29. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 70:24-71:21; Ex. 23, P-G5 at 67. Dr. Smith testified that the IRB exactly contains the S1 and S2 elements of the dispatch stack in

claims 1, 6, 14-15 and 18. That is, the IRB has a field for each source register specified by the instruction; in this case at least S1 and S2. Synopsizing his earlier testimony, Dr. Smith explained that (1) the source register specified by the instruction can be either an unrenamed or a rename register; (2) the S1 and S2 fields "specify where [*i.e.*, which register] the source values come from" for the instruction; and (3) the source register (*e.g.*, S1) specified by an instruction residing in the IRB is "whatever register is specified, as the instructions are residing in the IRB and data dependencies are being considered for that instruction."

30. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 71:23-72:21; Ex. 23, P-G5 at 67. Dr. Smith also testified that the IRB has elements that are at least equivalent to the S1 and S2 fields in the dispatch stack. Dr. Smith identified the function ("to specify the source"), way ("identifies the register number"), and result ("when the instruction launches to execute, it reads the correct value from the register") for the S1 and S2 fields. (72:3-8.) He explained why the corresponding source register fields in the IRB were identical or at least substantially identical in function/way/result to the S1 and S2 fields in the DS. Dr. Smith stated: "In the case of the IRB, the function of those fields is to identify a source operand value. The way it does that is it contains the register number for the renamed or the unrenamed register that contains the value, and the result is when the instruction launches, it will take the correct value from the source for execution and so you get the right answers." (72:8-14, emphasis added.) He indicated that his opinion was supported by his earlier testimony, including his discussion of the ERS and the schematics.

31. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 84:12-22; Ex. 23, P-G5 at 88-89. Based on ERS and the schematics and deposition testimony, Dr. Smith testified that the IRB has the dispatch stack's element of a field for each destination register specified by the instruction.

32. Ex. 12, 5/27/09 AM Tr. (Dkt. No. 1039) at 88:17-91:8; Ex. 24, P-G8 at 3-5. With reference to an actual instruction set sitting in the dispatch stack, Mr. Lotz testified that agreed that renaming remaps (1) all of the destination registers from general registers (*e.g.*, F0) to rename resisters (*e.g.*, R0) and (2) remaps the source registers (S1 and S2) of dependent instructions form general registers to rename registers. In other words, a given source register specified by dependent instructions is now a rename register. But, it does not remove or eliminate essential date dependencies

33. Ex. 12, 5/27/09 AM Tr. (Dkt. No. 1039) at 93:20-93:6. Mr. Lotz testified that each source tag (*e.g.*, source1 tag) in the IRB corresponds to a particular slot number and that the slot numbers in turn correspond to a particular rename register.

34. Ex. 12, 5/27/09 AM Tr. (Dkt. No. 1039) at 93:20-96:3; Ex. 25, P-G9 at 12-17. Mr. Lotz acknowledged that, once renaming occurs, the destination register for a given instruction residing in the IRB changes form a general register to a rename register (95:6-13). He further acknowledged that the slot number and the rename register are really the same thing: "Yes, there is one-to-one mapping, yes." (95:22-24; compare slides 15 and 16-17 in P-G9.)

35. Ex. 12, 5/27/09 AM Tr. (Dkt. No. 1039) at 127:21-128:20; Ex. 23, P-G5 at 53. With reference to an actual instruction set sitting in the IRB, Mr. Lotz testified (*e.g.*, 128:4-11, 128:17-129:1) that the source register for an instruction can be either a general register or a rename register. But, in either case, the same value is going to be supplied to the instruction. Thus, as an example, Mr. Lotz testified that the instruction in slot number one will supply its computed result, which is stored in rename register (R1), to the instruction (*i.e.*, instruction number 3) in slot number 2. (Note that the S1 register for the instruction in slot number 2 is rename register, R1.) Once the instruction in slot number one is retired, then its computed result is transferred from rename register (R1) to a general (*i.e.*, unrenamed) register. In any event, the same value will be supplied to instruction in slot number 2.

36. Ex. 11, 05/23/08 PM Tr. (Dkt. No. 1037) at 222:10-22; Ex. 28, DDX-51. Mr. Lesartre was shown an HP Exhibit (DDX-51) depicting four instructions in the IRB, with the slot number (*e.g.*, “slot 1”) written over certain source register (*e.g.*, S1 or S2) and over destinations register. Mr. Lesartre testified that, in the IRB, the slot numbers “are effectively the rename registers where we put our results.” In other words, the inference is that each slot number corresponds to a particular rename register and the source register specified by an instruction can be a rename register

37. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 213:9-13. Mr. Wheeler agreed that “an essential data dependency in the IRB, for that matter any instruction issuance unit, is where an instruction sitting in the IRB needs the computed result from a prior instruction before the instruction can issue.”

38. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 55:9-57:9; Ex. 22, P-G3 at 47-50. Dr. Torng explained that the α(S1) and α(S2) fields in the dispatch stack have logic that holds a value of one in the α(S1) or α(S2) fields which represents that there is an essential data dependency and then counts down (*i.e.*, decrements) that count to a value of zero when there is not any data dependency. He explained that the logic decrements a value greater than zero. He indicated that a count of one essential data dependency in, for example, the α(S1) field also indicates that there is one preceding instruction that will write its result to a destination register from which a subsequent instruction will get its operand (a/k/a value).

39. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 43:18-45:25; Ex. 23, P-G5 at 40. Dr. Smith explained that, when the dispatch stack is used with register renaming, the α(Si) field corresponds to a count of the number of essential data dependencies associated with a particular source register of the instruction and that this count is either a one or a zero. Stated another way, when the dispatch stack is implemented with renaming, the definition of α(Si) [*i.e.*, “the number of times that a particular register Si is used as a destination register in preceding

uncompleted instructions] corresponds to the number of essential data dependencies (*i.e.*, a count of one or zero). Specifically, Dr. Smith explained that with renaming, the maximum number of preceding uncompleted instructions that can write a result to a particular destination register (*e.g.*, rename register, R1) is at most one, since there can be at most one essential data dependency. So, the number of time that a source register (*e.g.*, the R1 source register in instruction 3 in slide 40) is used as a destination register in a preceding uncompleted instruction (*e.g.*, the R1 destination register in instruction 2 in slide 40) is one or zero (*see* the α(S1) field in slide 40). As Dr. Smith explained, a byproduct of the renaming is that the "that tracker" (*i.e.*, the α(S1) or α(S2) field corresponds to a count of the essential data dependencies, with the count being either one or zero.

40. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 57:18-59:23; Ex. 23, P-G5 at 50-52. Dr. Smith explained that each slot in the IRB has a "control block" that tracks essential data dependencies in the instructions. He indicate that each control block includes at least two essential dependencies trackers, which Dr. Swartzlander calls S1D and S2D "flags." Dr. Smith showed that the S1D flag performs the function of tracking essential dependencies associated with the S1 register in an instruction, while the S2D flag performs the same function with respect to dependencies associated with the S2 register in instructions. He explained that, when one of these "flags" contains a one, then that "indicates that there is one outstanding essential data dependency."

41. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 59:15-64:17, 65:5-18, 65:24-67:1; Ex. 23, P-G5 at 52-55. Using a sample set of instructions loaded in the IRB, Dr. Smith went through a detailed example of how the IRB detects whether instructions are free of data dependencies and works with the IRB's arbitration logic to simultaneously issue (a/k/a launch) multiple and out of order instructions per clock cycle that the IRB has determined are free of essential data dependencies (*e.g.*, shown by the two out of order instructions with the blue arrows in slide 53). Among other things, he explained that (1) the specified source registers

(S1 and S2) for the instructions can be unrenamed or renamed registers (61:7-62:15; 63:24-64:17); (2) the slot numbers in the IRB correspond to a particular rename register (*e.g.*, slot number one corresponds to rename register, R1 (61:7-13)); (3) the S1D and S2D flags in the IRB performed the recited function and delineated the recited function (62:3-9 and 63:6-13); (4) a value of one in a given flag indicate that there is a one essential data dependency and a value of zero indicates that there are not any essential data dependencies (61:19-63:13); (5) the value of one in each such flag is decremented to zero with the result that the instruction is free to launch as long as there is an available functional unit (*id.*); and (6) the specified destination registers (D) for instructions residing in the IRB are rename registers (61:7-18). Later, Dr. Smith further reiterated that "the IRB initially determines if instructions are free of data dependencies." (65:24-66:8 and P-G5 slides 53-55.) Then, with reference to slides depicting an IRB, he explained that such dependency free instructions ("that's indicated by these little arrows") and the instructions are simultaneously issued to the functional units with the arbitration logic. (*Id.*) Signals read the AL indicating that the instructions are "ready" to issue. Thus, Dr. Smith determined the recited detecting function and issuance (a/k/a/ launch) function.

42. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 69:15-18; Ex. 23, P-G5 at 62-63. Dr. Smith testified that the "logic" recited in the Court's claim construction for the dispatch stack "goes with" each of the essential dependency fields (α(S1) and α(S2)) and that logic is electronic circuitry.

43. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 73:10-15. Dr. Smith again made clear that each α(S1) and α(S2) essential dependency field in the DS has the "logic" recited in the Court's claim construction.

44. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 72:22-73:25. Dr Smith testified that the S1D and S2D flags in the IRB with their accompanying logic corresponded to the α(S1) and α(S2) essential dependency fields with their accompanying logic. As a recapitulation of his

prior discussion, he explained, by way of example, that the α(S1) essential dependency field holds a count of the number of data dependencies associated with the instruction and that this count can be either a zero or a one.

45. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 74:12-83:12; Ex. 23, P-G5 at 68-89. With reference to the ERS (P-355) and schematics (P-790) and his prior testimony, Dr Smith testified that the S1D and S2D flags in the IRB with their accompanying logic corresponded to the α(S1) and α(S2) essential dependency fields with their accompanying logic. He indicated that, in the IRB, each such "flag" is a hardware component, called a sclat0_4 latch, which holds a value of zero or one. He explained that these values correspond to a count of the number of essential data dependencies (either one or zero). He explained in detail how each such latch has associated logic which is the same as the logic set forth in the claim language for the α(S1) and α(S2) essential dependency fields in the dispatch stack, including the feature of decrementing a value >0 in each essential dependency field (*i.e.*, each S1D and S2D flag). He explained that the count of essential data dependencies (*i.e.*, zero or one) in turn represents "a comparison to see if the launching instruction register matches the register we're waiting for." (82:13-20.) In other words, it also represents the number of times (either zero or one) that a particular source register (*e.g.*, S1) is used as a destination register in a preceding instruction.

46. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 83:13-84:11; Ex. 23, P-G5 at 68-89. Dr. Smith explained there was "no substantial difference" between the α(S1) and α(S2) fields in the dispatch stack and S1D and S2D "flags" in the IRB. He identified the function ("track essential dependencies"), way ("it keeps a count of the outstanding essential dependencies, the ones yet to be resolved. As those are cleared, the count decrements when it reaches zero indicates no essential dependencies"), result ("the instruction is free to issue and it will provide the correct answers") for the α(S1) and α(S2) fields. (83:21-64:3.) He explained why the corresponding S1D and S2D "flags" were identical or at least substantially identical in function/way/result to the α(S1) and α(S2) with accompanying logic. Dr. Smith stated: "With

respect to the IRB, the function is to track the essential dependencies, and it does this by [way] having the flags we just identified hold a count of the essential dependencies, and outstanding ones, either one or zero, and as the dependencies resolve, those counts decrement to zero. The result is that when they hit zeros, we issue the instruction, it's free of data dependencies and we get the correct answer." (84:4-11, emphasis added.) He at least implicitly relied on his prior testimony as support for his conclusion, from which the jury reasonably concluded there was no substantial difference between such "flags" and the α(S1) and α(S2) fields.

47. Ex. 9, 05/22/08 PM Tr. (Dkt. No. 1034) at 160:24-161:3. Dr. Smith testified that a value of one in the α(S1) field means that there is a single essential data dependency, which in turn means that "there's one preceding instruction that writes to that register." In other words, Dr. Smith indicated that an essential dependency count of one indicates that there is one time that a given instruction's S1 register is used a destination register in a preceding uncompleted instruction.

48. Ex. 9, 05/22/08 PM Tr. (Dkt. No. 1035) at 186:6-11, 186:21-187:8, 162:14-20. Dr. Smith testified that α(S1) "represents the number of times that previous instructions write to that [source-S1] register." He explained (187:5-6) that, in the IRB of the PA-8000 family processors, α(S1) constitutes a count of dependencies ("in the case of the PA8000 I think they are"). When referred again to the definition of α(S1), Dr. Smith testified (162:14-20) that "in the case of PA-8000" α(S1) is a count of the number of essential data dependencies.

49. Ex. 16, 05/29/08 PM Tr. (Dkt. No. 1043) at 187:14-188:2, 162:16-20. Dr. Smith explained that he had considered the Court's definitions of α(S1) and α(S2) and that each of these definitions is indicative of a count of essential dependencies and that such count can be either one or zero. He reiterated that this was so in light of the use of register renaming (*see, e.g.*, Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 43:18-45:25 and P-G5, slide 40 above): the α(S1) and α(S2) count the number of, for a given register, S of I, count the number of previous

instructions that write to that register. We've been calling those essential dependencies, where one writes, the other reads. And it's a count. In the presence of register renaming, that count is either 0 or 1."

50. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 219:14-220:17; Ex. 3, P-34. Dr. Swartzlander testified that each α(S1) and α(S2) essential dependency field is the dispatch stack as construed by the Court has logic (a/k/a circuitry) associated with it that decrements a value greater than zero in the essential dependency field and determines when the value in that field is zero. He acknowledged that, when such value is zero, the instruction can issue as long as there is an available functional unit.

51. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 221:20-223:14; Ex. 25, P-G9 at 8. Dr. Swartzlander discussed a slide (no. 8) similar to a slide (Ex. 23, P-G5 at slide 53) that Dr. Smith had used to explain the IRB. The slide depicted the elements of the IRB, including the essential dependency fields (a/k/a S1D and S2D "flags') with a sequence of instructions. Dr. Swartzlander testified that the S1D and S2D flags detect whether there are any essential data dependencies. Dr. Swartzlander was then questioned about two instructions in the representative IRB that had an essential data dependency as denoted by the value of one for the S2D flag: namely, the instruction in slot zero (*i.e.*, the preceding instruction) and the instruction in slot 2 (note that the specified source register for S2 is rename register R0 which is the same register that the instruction in slot zero will write its computed result to). He acknowledged that (1) each such flag identifies the number of times that a renamed source register (*e.g.*, R1) is used as a destination register (*i.e.*, R1) in a preceding uncompleted instruction; (2) that in light of renaming such "number of times" is never going to be greater than one; and (3) such "number of time" corresponds to the quantity of essential dependencies. In other words, just like Dr. Smith testified, each S1D and S2D flag, not only conforms to the "definitions" of

α(S1) and α(S2),[1] but constitutes a count (either one or zero) of the number of data dependencies.

52. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 223:21-225:6. Dr. Swartzlander testified (223:22-224:1) that the IRB has latches (called sclat0_4 latches) that detect essential data dependencies in computer instructions stored in the IRB. He claimed that (224:16-19) that each such latch holds a "flag," rather than holding a "count," and therefore the latch does not count essential data dependencies. [Dr. Swartzlander referred to the latches (a/k/a essential data dependency detectors) as "S1D flags" and "S2D flags" to denote the particular source register (*e.g.*, S1) of the instruction with which the flag is associated. (*See* ¶ 40 herein at 58:22-59:6.)] Dr. Swartzlander claimed that he was surely aware of how the latch worked, based on his prior review of a particular schematic. (224:2-8 and 20-22.) He was then impeached with prior deposition testimony (Ex. 20, Swartzlander Dep. Tr. 188:8-11) in which he was shown this schematic, but testified that he was not sure how such latch worked.

53. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 225:16-226:8. Dr. Swartzlander testified that the definition of a count in computer architecture is the same as the definition of a count from other perspectives; namely a number that identifies the quantity of something. Thus, he acknowledged that the number one is a count, the number zero is a count and a count down from one to zero is a count.

54. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 226:8-16. Dr. Swartzlander testified that "a latch can hold a value that is interpreted as a count" and that one can implement a counter even with a single latch.

[1] Definition of **α(S1)**: "α(S1) is the number of times that an instruction's S1 register is used as a destination register in preceding, uncompleted instructions." (Ex. 36, Dkt. No. 1057, Claim Construction Jury Handout.) Definition of **α(S2)**: "α(S2) is the number of times that an instruction's S2 register is used as a destination register in preceding, uncompleted instructions." (*Id.*)

55. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 226:19-227:25; Ex. 25, P-G9 at 8. Dr. Swartzlander again discussed a slide (no. 8) similar to a slide (Ex. 23, P-G5 at slide 53) that Dr. Smith had used to explain the IRB. He admitted that a value of one in an S1D or S2D flag in the IRB indicates that there is a single essential data dependency associated with that flag; noting that the dependency was "between the rename registers" (*i.e.*, the computed result in a preceding instruction would be written to a rename register (*e.g.*, R1) and then supplied to the source register (R1) specified for a subsequent instruction – an essential data dependency among two instructions). Dr. Swartzlander was then impeached with prior deposition testimony (Ex. 20, Swartzlander Dep. Tr. 213:20-214:2) where he testified that when an S1D or S2D flag goes from a true (*i.e.*, a value of one) to a false (*i.e.*, a values of zero), "we've gone from having an essential data dependency to not having one."

56. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 228:12-24; Ex. 25, P-G9 at 9-10. Dr. Swartzlander testified that the α(S1) and α(S2) essential dependency fields in the dispatch stack are essential data dependency detectors, each of which decrements a value greater than zero, such as one, counts down to zero, and then the instruction is ready to "go." (*i.e.*, issue). He testified that each S1D and S2D flag in the IRB has a sclat0_4 latch and that there are over 56 such latches in the IRB.

57. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 86:1-88:16; Ex. 24, P-G8 at 1-2. With reference to a dispatch stack containing the sample instruction sequence shown in the patent, Mr. Lotz testified that a value of one in an essential data dependency field (*e.g.*, α(S1)) in the dispatch stack indicates that there is one essential data dependency among two instructions and agreed that this in turn "represents the number of time that this particular register here, S1, is used as a destination register in a prior instruction." (88:8-16.) In other words, for a dependency count of one or zero, the Court's definition of α(S1) or for that matter α(S2) corresponds to a count of the number of data dependencies, just like Dr. Smith testified.

58. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 101:21-102:2. Mr. Lotz testified that, when renaming is implemented in the dispatch stack, then the dependency count in any given essential dependency field [α(S1) or α(S2)] can at most be one. He agreed that the reason is that renaming remaps general register to rename registers.

59. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 102:10-103:14, 103:24-104:19, 105:1-13, 105:25-106:8; Ex. 25, P-G9 at 16-17. Mr. Lotz testified that each latch that embodies each S1D and S2D flag holds a value of one or zero (one = red flag and zero = green flag) and that a value of one indicates that there is one essential data dependency and a value of zero indicates that there are no essential dependencies. He acknowledged that when an essential dependency no longer exists the value in the latch changes form one to zero. He also acknowledged (102:23-103:18 and 106:4-8) that, in light of renaming, dependent instructions will get their source values from rename registers. In other words, the source register specified for a dependent instruction becomes a rename register.

60. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 113:9-19. Mr. Lotz effectively admitted that HP's use of the term flag is just another name for counting data dependencies. Mr. Lotz acknowledged that "the words HP uses to describe the process of where we have a count of one is a flag" and the words HP uses to "describe the process in the IRB where you have a count of zero … is the clearing of the flag."

61. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 125:23-126:3. Mr. Lotz acknowledged that, after renaming is performed on instructions, "the number of times that any given source register in a preceding uncompleted instruction that is still residing in the dispatch stack, can't be greater than one."

62. Ex. 11, 05/23/08 PM Tr. (Dkt. No. 1037) at 227:5-21. Mr. Lesartre testified that each sclat0_4 latch in the IRB contains a value of one when there is an essential data dependency and a value of zero when there are no such dependencies.

63. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 218:11-18, 219:2-19. Mr. Wheeler acknowledged that Mr. Tannenbaum wrote at least two non-liability opinions concerning the '115 patent. After testifying that he would disagree with Mr. Tannenbaum if Mr. Tannenbaum had indicated that a latch can be a one bit counter, he was presented with deposition testimony (Ex. 29, Tannenbaum Depo. Tr. 229:14-18) of Mr. Tannenbaum in which Mr. Tannenbaum testified that a latch can be a one bit counter.

64. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 43:18-45:25; Ex. 23, P-G5 at 50-52. Dr. Smith explained that each slot in the IRB has a "control block" that tracks essential data dependencies in the instructions. He indicate that each control block includes at least two essential dependencies trackers, which Dr. Swartzlander calls S1D and S2D "flags." Dr. Smith showed that the S1D flag performs the function of tracking essential dependencies associated with the S1 register in an instruction, while the S2D flag performs the same function with respect to dependencies associated with the S2 register in instructions. He explained that, when one of these "flags" contains a one, then that "indicates that there is one outstanding essential data dependency."

65. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 59:15-64:17, 65:5-18, 65:24-67:1; Ex. 23, P-G5 at 52-55. Using a sample set of instructions loaded in the IRB, Dr. Smith went through a detailed example of how the IRB detects whether instructions are free of data dependencies and works with the IRB's arbitration logic to simultaneously issue (a/k/a launch) multiple and out of order instructions per clock cycle that the IRB has determined are free of essential data dependencies (*e.g.*, shown by the two out of order instructions with the blue arrows in slide 53). Among other things, he explained that (1) the specified source registers (S1 and S2) for the instructions can be unrenamed or renamed registers (61:7-62:15; 63:24-64:17); (2) the slot numbers in the IRB correspond to a particular rename register (*e.g.*, slot number one corresponds to rename register, R1 (61:7-13)); (3) the S1D and S2D flags in the IRB performed the recited function and delineated the recited function (62:3-9 and 63:6-13); (4)

a value of one in a given flag indicate that there is a one essential data dependency and a value of zero indicates that there are not any essential data dependencies (61:19-63:13); (5) the value of one in each such flag is decremented to zero with the result that the instruction is free to launch as long as there is an available functional unit (*id.*); and (6) the specified destination registers (D) for instructions residing in the IRB are rename registers (61:7-18). Later, Dr. Smith further reiterated that "the IRB initially determines if instructions are free of data dependencies." (65:24-66:8 and P-G5 slides 53-55.) Then, with reference to slides depicting an IRB, he explained that such dependency free instructions ("that's indicated by these little arrows") and the instructions are simultaneously issued to the functional units with the arbitration logic. (*Id.*) Signals read the AL indicating that the instructions are "ready" to issue. Thus, Dr. Smith determined the recited detecting function and issuance (a/k/a/ launch) function.

66. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 73:4-15; Ex. 23, P-G5 at 68-69. Dr. Smith testified that he relied on the ERS (P-355) and the schematics in determining that the S1D and S2D "flags" in the IRB correspond to the α(S1) and αS2 essential dependency fields in the dispatch stack. Chapter 4 of the ERS (slide 69 and P-355) instructions are ready to issue (a/k/a launch) when they are dependency free.

67. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 219:14-220:24. Dr. Swartzlander testified that each α(S1) and α(S2) essential dependency field is the dispatch stack as construed by the Court has logic (a/k/a circuitry) associated with it that decrements a value greater than zero in the essential dependency field and determines when the value in that field is zero. He acknowledged that, when such value is zero, the instruction can issue as long as there is an available functional unit.

68. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 228:12-25, 229:16; Ex. 25, P-G9 at 9-10. Dr. Swartzlander testified that the α(S1) and α(S2) essential dependency fields are essential data dependency detectors, each of which decrements a value greater than zero, such as one,

counts down to zero, and then the instruction is ready to "go" (*i.e.*, issue). He testified that each S1D and S2D flag in the IRB has a sclat0_4 latch and that there are over 56 such latches in the IRB.

69. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 230:3-231:5, 232:1-4, 232:23-233:18; Ex. 25, P-G9 at 12-17. Dr. Swartzlander testified that, once instructions are inserted into the IRB, the destination registers in the instructions are changed from general registers to rename registers (*e.g.*, R3 to RR1) as long as the instructions are still in the IRB. With reference to slides 16-17, he acknowledged that this in turn causes the source registers (*e.g.*, S1) in dependent instructions to be changed from general registers to rename registers (*e.g.*, the S1 register in instruction number two is now rename register, RR1, because instruction number 1 is going to write its computed result (value) in rename register, RR1, and supply that value to instruction number 2 – *i.e.*, one essential data dependency exists). He also acknowledged (230:15-20 and slides 14-15) that each of the slot numbers in the IRB correspond to a unique rename register and vice versa.

70. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 77:9-16. Mr. Lotz testified that both the IRB and the DS detect whether instructions have any essential data dependencies

71. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 84:9-13. Mr. Lotz testified that the IRB "absolutely" has to efficiently detect whether instructions have any data dependencies in order to do its job of quick and efficient multiple and out of order instruction issuance.

72. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 107:19-24. Mr. Lotz testified that, in order to be able to launch instructions from the IRB, the IRB needs to detect whether the instructions have any data dependencies. He explained that "launch" means issue.

73. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 9:11-10:1 and 10:3-5. Mr. Lesartre testified that when instructions in the IRB are free of data dependencies, the instructions can be

launched (a/k/a issued) to available functional units which perform the appropriate operation (*e.g.*, Add) on the instruction. Mr. Lesartre further testified that, in the IRB for the PA-8000 family processors, the logic that launches (a/k/a issues) an instruction is called launch arbitration logic.

74. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 212:17-24, 212:25-213:5. Mr. Wheeler testified (212:17-24) that the IRB in the PA-8000 family processors detects whether there are any data dependencies in computer instructions. He further testified (212:25-213:5) that, if it were not for the IRB's ability to do so, the PA-8000 family processors would not be able to issue multiple and out of order instructions. The import of this testimony is that the IRB determines whether multiple and out of order instructions are data dependency free and ready to issue.

75. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 214:10-25. Mr. Wheeler testified that the IRB in the PA-8000 family processors has latches that determine whether instructions have any data dependencies. He classified these latches as detectors. The import of this testimony is that what HP calls the S1D and S2D flags in the IRB detect whether instructions are free of data dependencies.

76. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 229:1-5. Mr. Wheeler acknowledged that the IRB is the heart of the PA-8000 family processors and is responsible "for the detection of whether or not there are data dependencies in computer instructions."

77. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 24:9-27:19; Ex. 22, P-G3 at 24-29. Dr. Torng explained that a data dependency among instructions exists where a "younger" (*i.e.*, subsequent) instruction depends on an older (*i.e.*, preceding) instruction. (24:9-25:13.) He testified that there are two kinds of data dependencies: (1) essential and (2) non-essential and that there is no any technique that can remove essential data dependencies. (25:14-26:5 and 27:9-19.)

78. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 31:10-34:8; Ex. 22, P-G3 at 31-32. Dr. Torng explained register renaming in the context of an actual set of instructions whose values (*i.e.*, operands) can be stored in general registers (F0 through F6) or rename register (R0 through R6). He showed how renaming results in destination register of each original (unrenamed) instruction being renamed from a general register (*e.g.*, F5) to a renamed register (*i.e.*, R0). He showed that this in turn causes source registers in the original (unrenamed) instructions to be renamed for a general register to the appropriate rename register. It is evident from a comparison of the unrenamed and renamed instruction sets (P-G3 at slide 32) indicates that, after renaming, there is at most one time in which a source register is a destination register in a preceding uncompleted instruction and that this corresponds to a count of the essential dependency (*i.e.*, one or zero). Dr. Torng testified that renaming cannot remove or get rid of essential data dependencies in the instructions.

79. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 35:8-12. Dr. Torng testified that register renaming does not eliminate or remove essential data dependencies.

80. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 44:23-45:24; Ex. 22, P-G3 at 38-39. Dr. Torng testified that the number of times that his data dependency detection technique is implemented depends on the properties of the instructions, notably the instruction format and the types of data dependencies.

81. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 44:11-46:20; Ex, 22, P-G3 at 37-39. Dr. Torng testified that renaming does not detect whether a computer instruction has an essential data dependency. So, for example, he showed how his technique is implemented twice for instructions in an OP S1 S2 D format with no non-essential data dependencies and three times for such instruction in and OP S1 S2 S3 D format.

82. Ex. 4, 05/20/08 AM Tr. (Dkt. No. 1030) at 54:19-55:5. Dr. Torng testified that register renaming and his invention are two different things. He explained that his invention

detects whether instructions have any data dependencies, especially essential data dependencies, and that renaming at most removes non-essential data dependencies.

83. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 132:15-133:5. Dr. Torng testified that register renaming cannot detect essential data dependencies in computer instruction, but can remove or eliminate only non-essential data dependencies. Thus, even a processor that uses register renaming still needs to detect whether there are essential data dependencies among instructions in the processor.

84. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 13:4-16:15. Dr. Smith testified that the '115 patent covers a specific data dependency detection technique that allows a processor to do efficiently practical multiple out of order instruction issuance.

85. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 32:16-33:9. Dr. Smith testified that renaming does not eliminate or otherwise affect essential data dependencies in computer instructions. So, as to the PA-8000 family processors which implement renaming, the IRB still has to track such dependencies.

86. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 35:10-13. Dr. Smith testified that the Dr. Torng's invention as contained in the dispatch stack can be used with register renaming.

87. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 38:7-25. Dr. Smith testified that register renaming cannot remove essential data dependencies, although it can remove non-essential data dependencies.

88. Ex. 16, 05/29/08 PM Tr. (Dkt. No. 1043) at 203:8-208:17; Ex. 26, P-G11 at 70. With reference to a chart (slide 70), Dr. Smith testified in detail that the Torng patent contains a written description of each and every element for the dispatch stack that is set forth in claims 1, 6, 14-15 and 18.

89. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 42:20-43:4; Ex. 23, P-G5 at 39-40. Dr. Smith testified that if the dispatch stack is implemented in a processor that uses register renaming, "there is no reason to have a Beta D field" and that "it's gone, because there are no more nonessential data dependencies."

90. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 221:4-15. Dr. Swartzlander testified that Dr. Torng's invention can be used with register renaming and that there was not anything in the '115 patent that indicated otherwise. Further, he testified that, when Dr. Torng's invention is used with register renaming, the dispatch stack detects the essential data dependencies and register renaming removes non-essential data dependencies.

91. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 77:21-24. Mr. Lotz testified that register renaming as implemented in the IRB does not remove essential data dependencies.

92. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 78:6-9. Mr. Lotz testified that the Torng patent does not indicate that register renaming cannot be used with the dispatch stack.

93. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 91:25-92:8. Mr. Lotz acknowledged that any experienced processor designer "worth his salt" would not put any hardware in a processor to track non-essential data dependencies if he knew that the processor was going to be implemented with renaming that would remove all of the non-essential data dependencies.

94. Ex. 5, 05/20/08 PM Tr. (Dkt. No. 1031) at 213:9-214:8. Mr. Wheeler testified that essential data dependencies cannot be removed from computer instructions and that register renaming cannot detect whether instructions have any essential data dependencies.

95. Ex. 8, 05/22/08 AM Tr. (Dkt. No. 1034) at 100:20-101:9; Ex. 23, P-G5 at 92, 109. Dr. Smith concluded and amplified his testimony that the PA-8000 family processors infringe claims 1, 6, 14-15 and 18 of the '115 patent with an animation showing a side-by-side comparison of the dispatch stack and reservation circuit with the IRB and its accompanying

logic. Slide 92 captures an animation depicting that both the dispatch stack and IRB detect multiple instructions free of essential dependencies (green colored balls) and that such instructions are issued simultaneously and out of order by the reservation circuit and the arbitration logic. Slide 92 explicitly states: (1) "Multiple Instructions Issue at Once"; and (2) "Instructions Can Issue Out of Order."

96. Ex. 7, 05/21/08 PM Tr. (Dkt. No. 1033) at 220:24-221:1. Dr. Worley testified that register renaming cannot issue essential data dependencies in computer instructions.

97. Ex. 7, 05/21/08 PM Tr. (Dkt. No. 1033) at 224-14-17. Dr. Worley acknowledged that the IRB in the PA-8000 family processors still needs to detect essential data dependencies even when the IRB is implemented with renaming.

98. Ex. 12, 05/27/08 AM Tr. (Dkt. No. 1039) at 91:1-8. Mr. Lotz testified that, because register renaming cannot eliminate essential data dependencies, they need to be detected.

99. Ex. 7, 05/21/08 PM Tr. (Dkt. No. 1033) at 253:4-7, 220:12-17. Dr. Worley acknowledged (253:4-7) that register renaming does not eliminate the need to detect essential data dependencies (a/k/a, "read-after-write dependencies"). Dr. Worley also acknowledges (220:12-17) that a "read-after-write" data dependency is also known as an essential data dependency.

100. Ex. 7, 05/21/08 PM Tr. (Dkt. No. 1033) at 256:8-12. Dr. Worley testified that, even if register renaming eliminates every single non-essential data dependency in the IRB of the PA-8000 family processors, the IRB still has to detect whether the instructions have any essential data dependencies.

101. Ex. 7, 05/21/08 PM Tr. (Dkt. No. 1033) at 256:8-17. Dr. Worley testified that the IRB must have a data dependency detector associated with each field of the instruction.

102. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 205:24-206:4. Dr. Swartzlander acknowledged that Dr. Smith is "one of the foremost experts in the world" in the very field relevant to this case; namely, computer architecture and particularly in the field of high performance computing.

103. Ex. 13, 05/27/08 PM Tr. (Dkt. No. 1040) at 203:14-19, 204:16-25, 205:14-18, 206:8-14, 207:13-208:3. Dr Swartzlander testified (203:14-19) the lawsuit was about high performance computers, particularly high performance processors and computer speed was an important factor. He acknowledged that his evaluation of computer performance over the years was based in large part on his own personal use of computers, just like the rest of the public. He admitted (204:16-25; 205:14-18) that the only evaluation of high performance computers he had done was in the 1980s and none of those computers were capable of multiple out of order instruction issuance. He could not point to (207:13-208:3) any instances of evaluation of multiple and out of order issuance processors, apart from his work on this lawsuit. Accordingly, the evidence supported an inference by the jury that Dr. Swartzlander's experience was modest at best.

104. Ex. 17, 05/30/08 Tr. (Dkt. No. 17) at 31:1-35:25, 96:1-12; Ex. 27, Closing Slides at 50-64. At closing argument, counsel for Cornell provided linking argument on infringement under the doctrine of equivalents. Regarding S1 and S2, counsel, while referring to slide 54, explained that the IRB in the PA-8000 family processors has a field that calls out the "container" (*i.e.*, register) from which the instructions receive their source values (34:8-11). Slide 54 shows how the S1 and S2 fields in the IRB refer to either general or rename registers as their sources.

105. Ex. 17, 05/30/08 Tr. (Dkt. No. 17) at 31:1-35:25, 96:1-12; Ex. 27, Closing Slides at 50-64. Furthermore, counsel for Cornell provided linking argument on the α(S1) and α(S2) fields. Counsel, referring to slides 55-58, explained that the IRB contains fields for detecting

essential dependencies (34:11-13). These fields, or "flags" in HP's parlance, hold counts that represent the number of essential data dependencies (34:13-18). Counsel also linked the definition of α(S1) and α(S2) to the context of general versus renamed registers in the IRB in the PA-8000 family (33:23-25, 35:20-25). Specifically, counsel compared α(S1) and α(S2) in instruction streams without and with renaming and demonstrated that with renaming the number of times that a source register in an instruction refers to the destination register of a preceding, uncompleted instruction is at most 1 (*id.*, *see also* slides 61-62). Accordingly, the α(S1) and α(S2) fields (*i.e.*, "flags") receive initial values of "1" if a dependency is present which is subsequently decremented to "0" (34:16-25). Counsel further directed the jury's attention to slide 60 as demonstrating how the IRB reads on the Court's construction of α(S1) and α(S2) (96:1-12). The jury was directed specifically to the Court's definition of α(S1) and α(S2) repeated verbatim on slide 60, which further states, pointing to a α(S1) field (*i.e.*, "flag"): "Number of times RR1 is Written in Preceding, Uncompleted Instructions."

106. Ex. 13, 5/27/08 PM Tr. (Dkt. No. 1040) at 203:20-204:8. Dr. Swartzlander testified that Dr. Torng's invention boosts the performance of computers by enabling them to efficiently execute multiple and out-of-order instructions. Dr. Swartzlander further testified that Dr. Torng's invention efficiently detects data dependencies among computer instructions.

107. Ex. 5, 05-20-08 PM Trial Tr. (Dkt. No. 1031) at 146:8-12, 152:15-153:20. John Wheeler, the manager of the team that developed the PA-8000 processor, testified that HP is in the business of selling servers and workstations, and that HP stated that processors are the heart of the systems as reflected in P-734 at page 4.

108. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 9:5-10:20, 14:24-15:17, 76:17-77:16. Robert Noller[2] testified concerning a 1995 Q&A document (P-232 at HP 262008) which

[2] Mr. Noller joined the Microprocessor Group in 1993 as a marketing engineer leading competitive efforts. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 5:17-20.

states HP does not sell microprocessors on the open market.[3] Mr. Noller also testified that HP did not create business plans for processors since HP does not sell processors and did not report profits or losses on processors. Mr. Noller also testified HP is not in the business of selling microprocessors and did not sell stand alone processors.

109. Ex. 8, 05-22-08 Trial AM Tr. (Dkt. No. 1034) at 101:16-118:17; Ex. 9, 05-22-08 PM Tr. (Dkt. No. 1035) at 149:19-151:5, 153:19-154:5. Dr. James Smith testified as a qualified expert concerning the performance benefit provided by the ’115 invention to systems using the PA-8000 series of CPUs. The Torng invention provides on average at least an 84% improvement in floating point performance, and on average at least a 50% benefit for integer performance. The floating point performance improvement would benefit HP customers such as Boeing and Ford. By using the Torng invention, HP systems built on PA-8000 series processors were the leaders in performance. Without the Torng invention, those HP systems would have trailed the competition in terms of performance. The data and documentation which Dr. Smith considered and/or relied upon in presenting his opinions, as identified in P-G5, included P-44, P-82, P-307, P-315, P-355, P-540, P-627, P-766, P-1441, and P-1567.

110. Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 271:11-297:4; Ex. 8, 05-22-08 AM Trial Tr. (Dkt. No. 1034) at 9:21-13:3; 36:7-37:9; Ex. 16, 05-29-08 PM Trial Tr. (Dkt. No. 1043) at 188:3-198:25, 210:12-214:4. Dr. James Smith testified as a qualified expert that the Torng patent provided a breakthrough solution to a longstanding problem preventing the efficient issuance of multiple instructions out-of-order, in contrast to the inefficient and impractical “brute force” approach discussed in the prior art, including the Hasbrouck patent (P-6). The data and documentation which Dr. Smith considered and/or relied upon in presenting his opinions, as identified in P-G5 and P-G11, included P-1 and P-6, as well as DDX-4.

[3] *See also* P-135, HP 262002-9.

111. Ex. 8, 05-22-08 AM Trial Tr. (Dkt. No. 1034) at 22:17-22. Dr. James Smith also opined that the PA-8000 series processors were critical to HP's servers and workstations, since those systems have no use without the processor.

112. Ex. 13, 05-27-08 PM Trial Tr. (Dkt. No. 1040) at 203:20-204:8, 219:2-13; Ex. 20, Swartzlander Deposition Clip at 382:7-24. HP retained expert Earl Swartzlander confirmed that the dispatch stack invention relates to high-performance computers, and boosts the performance of such systems by enabling efficient multiple and out-of-order execution of instructions. At his deposition, Dr. Swartzlander opined that the performance benefit from out-of-order execution was between 25% to 55%.

113. Ex. 5, 05-20-08 PM Trial Tr. (Dkt. No. 1031) at 150:4-9, 151:18-152:3, 212:12-214:6, 228:1-229:9. John Wheeler was questioned concerning an HP admission in an April 5, 2002 letter to the Court (Dkt. No. 14, P-1872) that "Indeed, when HP released the PA-8000 CPU in 1996, it was the world's fastest processor, based largely on its ability to issue multiple, out-of-order instructions per CPU clock cycle." Mr. Wheeler also testified that, if it were not for the IRB's ability to detect data dependencies, the PA-8000 could not issue multiple, out-of-order instructions and that register renaming cannot remove essential data dependencies. Mr. Wheeler also confirmed that HP's opinion counsel wrote concerning the PA-8000 (at page 4 of D-629) that the "out-of-order execution capability of the HP product allows it to attain peak superscalar performance by instruction execution as data dependencies are resolved." Mr. Wheeler also confirmed that the IRB is the very heart of the PA-8000 from a functionality control standpoint.

114. Ex. 12, 05-27-08 AM Trial Tr. (Dkt. No. 1039) at 13:4-11. Gregg Lesartre confirmed that the IRB can launch up to 4 instructions every cycle, as specified in the External Reference Specification (P-355 at chapter 5, page 5-8).

115. Ex. 12, 05-27-08 AM Trial Tr. (Dkt. No. 1039) at 26:25-27:14, 77:13-24, 80:2-7, 82:16-83:11, 84:20-85:1, 91:21-92:11, 95:25-96:3, 124:1-18. Jonathan Lotz testified that the initial goal of reordering instructions, so as to allow instructions to issue out of program order, was to solve the memory latency problem, by which instructions could be delayed from issuing while waiting on memory. (*See also* P-317 at AK 00038; P-150 at HP 602762.) Mr. Lotz also confirmed that register renaming cannot remove essential data dependencies, and that the register renaming used in the PA-8000 was only a little different from the prior art. Mr. Lotz also confirmed that the IRB is the central control logic for the entire processor. Mr. Lotz also agreed that additional functional units would be wasted absent robust dependency tracking that enables efficient multiple and out-of-order instruction issuance.

116. Ex. 5, 05-20-08 PM Trial Tr. (Dkt. No. 1031) at 157:5-158:12, 220:25-221:4. John Wheeler testified that he accepted the recommendation of Darius Tanksalvala to implement a processor with multiple and out-of-order instruction issuance capability in 1991 or 1992. Mr. Wheeler also confirmed that HP needed to introduce servers and workstations with PA-8000 processors into the marketplace in August, 1996 in order to remain competitive with the likes of IBM and Sun Microsystems.

117. Ex. 5, 05-20-08 PM Trial Tr. (Dkt. No. 1031) at 158:18-163:13; Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 171:1-179:10; Ex. 30, William Worley Deposition Designation at 185:9-12. In 1991, William Worley[4] sent an analysis to senior HP management concerning the direction HP should take on computer processors. In this analysis (P-766[5] at HP 014861), Mr. Worley discusses the significant challenge in implementing an out-of-order processor due to the "dependency jungle."

[4] Mr. Worley is a former HP employee retained by HP's counsel as a consultant in this case. Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 169:14-171:3.

[5] *See also* P-700.

118. Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 179:11-183:1, 183:14-23. William Worley also confirmed that the IRB is the very heart of a multiple and out-of-order issuance processor, as noted at page 24 of P-766. Also, superior product performance and cost were essential to leadership in the server and workstation markets, as noted at page 5 of P-766. *See also* id. at 284:16-286:16, 291:1-292:3.

119. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 26:24-29:20. Robert Noller confirmed that a September 4, 1992 HP planning document (P-441) states that even with PA-7000 and PA-7100, HP would fall behind IBM on performance in 1993 and 1994 (HP 041422), that PA-8000 was needed to regain performance leadership (HP 041421) and out-of-order execution would be a competitive enabler of the PA-8000 (HP 041422). *See also* P-437 at HP 041306 (August 27, 1992 HP business planning document recognizes that IBM was pressuring HP on performance, and PA-7000 family was performance deficient); P-1349 at HP 331925 (May 18, 1992 HP Systems Technology Division business planning document states that HP intends to maintain industry leading price/performance for PA-RISC).

120. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 10:21-11:21. Robert Noller also confirmed that an August 10, 1995 draft Q&A (P-232 at HP 262007), states that out of order execution was the only means known to HP to execute up to four instructions per clock cycle.[6] *See also* Ex. 10, 05-23-08 AM Trial Tr. (Dkt. No. 1036) at 106:15-24.

121. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 204:7-208:8; Ex. 18, Cox Deposition Designation at 39:3-18. Brian Cox, HP's worldwide director of product marketing, confirmed that an October 6, 1992 HP competitive assessment to HP senior management (P-132 at HP 329157) recognized that IBM was pressuring HP on performance, and HP would not regain clear performance leadership until launch of PA-8000. HP's UNIX marketing strategy was based on PA-RISC systems providing a decisive performance advantage over the

[6] *See also* P-135, HP 262002-9 at HP 262007

competition (HP 329158). In many instances potential customers use comparisons on industry benchmarks to form a short list of vendors to consider for evaluation, and HP followed a strategy of publishing industry benchmarks in an effort to increase name brand recognition (HP 329168). *See also* P-457 at HP 010185 (1993 Microprocessor Report states IBM takes performance leadership); P-1407 at HP 700007.0008 (November 3, 1993 HP management report states that HP will not be competitive until launch of PA-8000).

122. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 208:9-14. Brian Cox also confirmed that the failure of the PA-8000 series to remain performance competitive could have had an adverse financial impact on HP.

123. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 209:12-23. Brian Cox also testified that independent software vendors (ISVs) want the CPU used in systems for which they may write applications to remain performance competitive.

124. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 29:21-30:2; 34:22-36:18; Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 176:20-178:18; Ex. 21, Tanksalvala Deposition Designation at 114:22-25, 115:1-25, 116:5-9, 117:13-119:7, 119:13-22, 122:2-122:5, 122:11-18, 125:13-18, 128:10-15, 128:22-129:2, 130:5-8, 131:18-25, 133:7-19, 133:21-25, 138:16-20. Multiple witnesses confirmed the accuracy of the statement in the August 1993 Design Objective Sign Off ("DOSO") that out-of-order execution (dynamic scheduling) would be a competitive requirement (P-440 at HP 052479-80).[7] The DOSO also states (P-440 at HP 052523) that customer needs include "leadership application performance" and "highest performance RISC architecture." The DOSO also states (P-440 at HP 052472) that the PA-8000 will maintain performance leadership in high-end and mid-range computer systems. The DOSO also states (P-440 at HP 052528) that an HP marketing concern was "competitive performance" and "leadership performance" in benchmarks. *See also* P-734 at HP 056690-91

[7] *See also* P-307; P-282; and P-638.

(November 2001 VLSI Technology Operation report states "at the heart of each of HP's computer products is a microprocessor. . . . Microprocessor design is at the source of a value chain that ultimately delivers computing solutions to customers").

125. Ex. 17, 05-30-08 Trial Tr. (Dkt. No. 1044) at 42:19-43:1. As ordered by the Court (Dkt. No. 446 at 10 (P-36)): "It is established that, at least as of 1996, HP's customers for servers and workstations had received HP's press releases…Those press releases were intended to convey to HP's customers that the 'Intelligent Execution' feature discussed in them was at least one of the reasons why they should buy HP servers and workstations containing PA-8000 family processors." *See also* Dkt. No. 592 (P-38), Order Affirming Magistrate Judge's 09/26/2005 Preclusion Order at Dkt. No. 446.

126. Ex. 5, 05-20-08 PM Trial Tr. (Dkt. No. 1031) at 210:12-213:5; Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 67:2-10. HP's April 4, 1996 press release (P-167 at HP 041235) states "[i]ntelligent execution is based on the PA-8000's out-of-order execution capability."[8]

127. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 96:7-18, 116:3-117:24. Todd Gustafson confirmed that HP sent press releases to all industry contacts and PR Newswire for distribution. Mr. Gustafson also recalled that the PA-8000 was a big deal at HP because of the increased performance.

128. Ex. 5, 05-20-08 PM Trial Tr. (Dkt. No. 1031) at 147:3-149:20. John Wheeler testified that when the PA-8000 was introduced in HP servers and workstations in 1996, the multiple and out-of-order instruction execution capability of the PA-8000 was emphasized in HP marketing materials. HP emphasized the multiple and out-of-order instruction execution

[8] *See also* P-460, HP 041234-HP 041235.

capability of the PA-8000 to catch the eye of potential purchasers of HP servers. Mr. Wheeler also emphasized this in his own resume at P-696.

129. Ex. 8, 05-22-08 AM Trial Tr. (Dkt. No. 1034) at 26:21-31:2. HP articles, such as P-315 and P-627, describe the IRB as the conductor that runs the show in the PA-8000 CPU. The IRB is the heart of the machine, and provides out-of-order capability (P-315 at AK 00060). The IRB serves as the central control logic for the entire chip, even though it consumes only 20% of the die area (P-315 at HP AK 00063; P-627 at HP 006844). Adding functional units to the processor is not enough to improve performance, a robust dependency tracking system is necessary (P-627 at HP 006844).

130. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 5:25-6:13, 20:6-21:7. Robert Noller testified that HP would use Microprocessor Reports for benchmark results in competitive assessments, and would publish benchmark results in Microprocessor Reports.

131. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 26:16-23, 95:13-17. Robert Noller and Todd Gustafson testified that HP customers rely on benchmarks to qualify vendors on the "horsepower" systems could deliver.

132. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 84:23-86:22. Robert Noller confirmed that the paper on Memory Performance Features of the PA-8000 states (P-150 at HP 602762) that initial benchmark performance expectations for PA-8000 systems at introduction in early 1996 would be delivered by a "56 entry instruction reorder buffer enabling full out of order execution of code" and "a peak execution rate of 4 instructions per cycle."

133. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 46:25-47:20. Robert Noller met with and discussed out-of-order execution with HP customers.

134. Ex. 11, 05-23-08 PM Trial Tr. (Dkt. No. 1037) at 162:15-163:8. September 16, 1996 Business Wire article (P-1080, CR0317591-97 at CR0317591-2) states "For compute-

intensive tasks, the PA-8000 20.2 SPECfp95 rating is more than 40 percent faster than the nearest competitor in its class. Advanced optimizing compilers and 'intelligent execution' allow the PA-8000 to attain superscalar performance levels unmatched by other vendors."

135. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 88:23-89:17. Todd Gustafson[9] confirmed that, for customers of workstations performing finite element analysis such as crash testing, the processor's floating point performance was unequivocally critical.

136. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 93:7-96:6. Todd Gustafson confirmed that in 1996, HP workstation customers in the $20 billion technical computing market had an insatiable demand for performance, and were performance driven. Mr. Gustafson also confirmed that a 1996 HP presentation on Technical Computing (P-230) states that the new PA-8000-based systems provide leadership compute performance (HP 146650), that the PA-8000 was the most powerful microprocessor in the world (HP 146656), and quoted HP customer Fluent, Inc. about the importance of the performance delivered by the PA-8000 (HP 146657).

137. Ex. 13, 05-27-08 PM Trial Tr. (Dkt. No. 1040) at 236:16-238:16; Ex. 20, Swartzlander Deposition Designation at 316:9-13. The IEEE Micro article "HP PA-8000 RISC CPU" (P-622) was written by Ashok Kumar, one of the authors of the chapter describing the IRB in the External Reference Specification. Exhibit P-170 states at HP 000904 that the IRB is the heart of the machine and provides out of order execution capability.[10]

138. Ex. 14, 05-28-08 Trial Tr. (Dkt. No. 1041) at 116:12-120:17. A 1996 presentation by Ashok Kumar of HP at Hot Chips (P-317) states at AK 00026 that the IRB was the heart of the machine. HP's Hot Chips presentation states at AK 00027 that the primary

[9] Mr. Gustafson is a sales vice president with HP responsible for workstations. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 87:10-88:14, 90:6-11, 104:7-21.

[10] *See also* P-315, AK 00059-AK 00067.

design objective of the PA-8000 was to obtain industry leading performance. HP's Hot Chips presentation states at AK 00030 that the most notable feature of the PA-8000 is the IRB, which serves as the central control unit. HP's Hot Chips presentation states at AK 00032 that on chip cache only provided a 5% performance improvement, and the IRB allowed HP to hide the effects of cache latency. HP's Hot Chips presentation states at AK 00038 that the IRB is of paramount importance, and also leads to the solution for another bottleneck, memory latency. HP's Hot Chips presentation states at AK 00039 that the IRB serves as the central control logic for the entire chip, yet consumes less than 20% of the die area.

139. Ex. 10, 05-23-08 AM Trial Tr. (Dkt. No. 1036) at 41:8-45:7. 1999 customer survey (P-1148, HP 056599-HP 056635.27 at HP 056615) shows that 72% of customers would switch vendors due to poor application performance.

140. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 202:1-202:16; Ex. 18, Cox Deposition Designation at 122:19-21, 122:23, 122:25-123:1, 123:3-8, 123:10-12, 123:14-15. Brian Cox testified that he had seen customer surveys listing performance as the first listed criteria.

141. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 22:9-26:15. Robert Noller confirmed that a 2000 White Paper (P-436), prepared for use with HP customers, discussed the performance limitations imposed by data dependencies (HP 049719), the at least 30% performance advantage of the PA-8000 over the prior generation PA-7200 (H P049719), and the solution to the dependency problem HP used in the instruction reorder buffer of tracking dependencies so that up to four instructions may issue per clock cycle (HP 049721). *See also* P-174, HP 725535-.0018 at HP 725535.0006 (a 2000 HP White Paper emphasizes importance of dependency tracking to performance).

142. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 17:5-22:8. Robert Noller testified at his deposition that a 2001 White Paper (P-464) was created to educate HP sales and

marketing on the competitive advantages of the PA-8400. This white paper identifies the PA-8700 as the heart of the system (HP 040927), identifies the performance advantages of the PA-8700 over the competition (HP 040928) and references the "56-way instruction reorder buffer" as a performance attribute of the processor and system (HP 040929).

143. Ex. 6, 05-21-08 AM Trial Tr. (Dkt. No. 1032) at 96:19-98:19. Todd Gustafson confirmed that a December 17, 2002 Two-Minute Customer Overview concerning the J6750 workstation trained HP's sale force to stress the incredible ("smoking") processing performance based on the PA-8700+ processor.

144. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 246:1-252:9; Ex. 10, 05-23-08 AM Trial Tr. (Dkt. No. 1036) at 26:24-31:10. Dr. Marion Stewart testified as a qualified expert concerning the substantial customer demand for the patented features and benefits of the '115 patent. The data and documentation which Dr. Stewart considered and/or relied upon in presenting his opinions, as identified in P-G6, included P-132, P-135, P-170, P-174, P-178, P-236, P-307, P-437, P-440, P-441, P-457, P-460, P-463, P-466, P-733, P-1148, P-1349, P-1407, P-1501, P-1657.

145. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 219:8-229:22. Brian Cox[11] testified that HP sales technicians would select PA-8000 family CPU modules from the HP Order and Configuration Guides (D-240, D-637, D-638) in configuring customer orders for servers, clarifying on redirect that the components referenced in the order and configuration guides were CPU modules, and not mere processors.

146. Ex. 8, 05-22-08 AM Trial Tr. (Dkt. No. 1034) at 106:3-107:6. Dr. James Smith testified as a qualified expert that HP customers such as Boeing and Ford would make use of the performance benefit provided by the '115 claimed invention. In his testimony, Dr. Smith

[11] Mr. Cox is presently HP's worldwide director of product marketing. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 210:11-23.

referred to P-44, a June 4, 2007 Stipulation and Order concerning HP sales to Ford and Boeing of systems containing each generation of PA-8000 series CPUs, and the use by Ford and Boeing of systems containing each generation of PA-8000 series CPUs.

147. Ex. 14, 05-28-08 Trial Tr. (Dkt. No. 1041) at 211:16-213:6; Ex. 15, 05-29-08 AM Trial Tr. (Dkt. No. 1042) at 7:11-10:12, 28:4-35:19, 50:2-23. Robert Wallace, an accountant retained by HP, testified concerning his calculation of the $23 billion royalty base as defined by revenue from sales of CPUs as set forth in DDX-245 and DDX-246.[12] Mr. Wallace testified that there were in excess of 700,000 transactions related to CPUs as reflected in HP's business records. In contrast, Mr. Wallace could only estimate revenue attributable to processors since only 31,361 processors of 1,9000,000 were transferred separate from a CPU brick or system, and he lacked data for several generations of the PA-8000 series. Mr. Wallace was also aware that at least a portion of the 31,361 processors went to strategic partners with whom HP has a number of agreements concerning development and systems.

148. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 242:7-244:9; Ex. 10, 05-23-08 AM Trial Tr. (Dkt. No. 1036) at 31:14-33:19; 60:24-65:16. Dr Marion Stewart testified as a qualified expert concerning facts significant to the selection of a royalty base, including that nearly 99% of PA-8000 series processors were included in CPUs in systems sold by HP. Dr. Stewart also testified that the remaining 1.5% appear to have been transferred as part of strategic relationships or as CPU upgrades. The data and documentation which Dr. Stewart considered and/or relied upon in presenting his opinions, as identified in P-G6, included P-132,

[12] In contrast, the CPU profit and loss statement which HP created early in this litigation, as reflected at P-1657, estimated CPU revenues at $2.1 billion. This purported profit and loss statement was generated by HP prior to the December 16, 2004 evidentiary hearing discussed in the testimony of Mr. Gonzales (Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 141:6-141:22), and reflected in P-1824 (Dkt. No. 333) at 166:3-173:11. *See also* Docket No. 448 at 9-10, 15-25, in which the Court ordered HP to provide further financial discovery in light of evidence presented at the evidentiary hearing, including the Xpress and OSRM data relied upon by Mr. Wallace in his CPU revenue calculation of $23 billion.

P-135, P-170, P-174, P-178, P-236, P-307, P-437, P-440, P-441, P-457, P-460, P-463, P-466, P-733, P-1148, P-1349, P-1407, P-1501, and P-1657.

149. Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 157:5-161:10; Ex. 19, Michael Phelps Deposition Designation at 95:25-96:23. Michael Phelps[13] confirmed that P-545 is a six month review of workstation sales data which shows HP sold 8,722 workstation systems containing PA-8000 series processors, sold eight (8) PA-8000 series CPUs (CPU SKUs listed are AB630A, AB636A and AB637A in standalone details by product), and which does not identify any sales of stand alone PA-8000 series processors. Mr. Phelps testified that HP is not in the business of selling processors.

150. Ex. 14, 05-28-08 Trial Tr. (Dkt. No. 1041) at 90:13-16, 97:23-98:10, 102:24-103:6, 103:10-104:13, 104:20-105:3, 105:17-106:11, 109:11-112:8. Greg Huff testified that upon joining HP, he worked on the "development of servers based on PA RISC CPUs." Mr. Huff testified that he was familiar with the components on in the Superdome server, and was permitted to open a Superdome and describe those components to the jury. In describing the CPU brick, Mr. Huff testified that "when we sell a processor, this is what you get, this whole assembly." Mr. Huff also explained the heat sink and power supply of the CPU brick were necessary for the processor to operate. Mr. Huff also testified that ICOD as referenced at P-55 at HP 341120, under which customers could pay for the use of CPUs, was "essentially a CPU debit card."

151. Ex. 9, 05-22-08 PM Trial Tr. (Dkt. No. 1035) at 198:23-201:25. Brian Cox testified that HP does not sell microprocessors on the open market, and that limited sales of processors to OEMs were part of a broader strategic relationship to expand HP's market

[13] Mr. Phelps is a financial specialist with HP's workstation group, and appeared on two occasions as an HP corporate witness. Ex. 7, 05-21-08 PM Trial Tr. (Dkt. No. 1033) at 156:1-157:4.

penetration overseas and which included sales of systems, development agreements, and licenses. Such OEM system sales are referenced in P-154 at HP 271734.